Learning Regular Omega Languages

Dana Angluin\textsuperscript{1,*} and Dana Fisman\textsuperscript{2,**}

\textsuperscript{1} Yale University, New Haven, Connecticut, USA
\textsuperscript{2} University of Pennsylvania, Philadelphia, Pennsylvania, USA

Abstract. We provide an algorithm for learning an unknown regular set of infinite words, using membership and equivalence queries. Three variations of the algorithm learn three different canonical representations of omega regular languages, using the notion of families of DFAs. One is of size similar to $L$, a DFA representation recently learned using $L^\ast$ \cite{7}. The second is based on the syntactic forc, introduced in \cite{14}. The third is introduced herein. We show that the second can be exponentially smaller than the first, and the third is at most as large as the first two, with up to a quadratic saving with respect to the second.

1 Introduction

The $L^\ast$ algorithm learns an unknown regular language in polynomial time using membership and equivalence queries \cite{2}. It has proved useful in many areas including AI, neural networks, geometry, data mining, verification and many more. Some of these areas, in particular verification, call for an extension of the algorithm to regular $\omega$-languages, i.e. regular languages over infinite words.

Regular $\omega$-languages are the main means to model reactive systems and are used extensively in the theory and practice of formal verification and synthesis. The question of learning regular $\omega$-languages has several natural applications in this context. For instance, a major critique of reactive-system synthesis, the problem of synthesizing a reactive system from a given temporal logic formula, is that it shifts the problem of implementing a system that adheres to the specification in mind to formulating a temporal logic formula that expresses it. A potential customer of a computerized system may find it hard to specify his requirements by means of a temporal logic formula. Instead, he might find it easier to provide good and bad examples of ongoing behaviors (or computations) of the required system, or classify a given computation as good or bad — a classical scenario for interactive learning of an unknown language using membership and equivalence queries.

Another example, concerns compositional reasoning, a technique aimed to improve scalability of verification tools by reducing the original verification task into subproblems. The simplification is typically based on the assume-guarantee reasoning principles and requires identifying adequate environment assumptions.

* Research of this author was supported by US NSF grant CCF-0916389.
** Research of this author was supported by US NSF grant CCF-1138996.
for components. A recent approach to the automatic derivation of assumptions uses $L^*$ \cite{5, 11, 17} and a model checker for the different component playing the role of the teacher. Using $L^*$ allows learning only safety properties (a subset of $\omega$-regular properties that state that something bad hasn’t happened and can be expressed by automata on finite words). To learn liveness and fairness properties, we need to extend $L^*$ to the full class of regular $\omega$-languages — a problem considered open for many years \cite{11}.

The first issue confronted when extending to $\omega$-languages is how to cope with infinite words? Some finite representation is needed. There are two main approaches for that: one considers only finite prefixes of infinite computations and the other considers ultimately periodic words, i.e., words of the form $uv^\omega$ where $v^\omega$ stands for the infinite concatenation of $v$ to itself. It follows from McNaughton’s theorem \cite{15} that two $\omega$-regular languages are equivalent if they agree on the set of ultimately periodic words, justifying their use for representing examples.

Work by de la Higuera and Janodet \cite{6} gives positive results for polynomially learning in the limit safe regular $\omega$-languages from prefixes, and negative results for learning any strictly subsuming class of regular $\omega$-languages from prefixes. A regular $\omega$-language $L$ is safe if for all $w \notin L$ there exists a prefix $u$ of $w$ such that any extension of $u$ is not in $L$. This work is extended in \cite{8} to learning bi-$\omega$ languages from subwords.

Saoudi and Yokomori \cite{19} consider ultimately periodic words and provide an algorithm for learning in the limit the class of local $\omega$-languages and what they call recognizable $\omega$-languages. An $\omega$-language is said to be local if there exist $I \subseteq \Sigma$ and $C \subseteq \Sigma^2$ such that $L = I \Sigma^\omega - \Sigma^* C \Sigma^\omega$. An $\omega$-language is referred to as recognizable \cite{19} if it is recognizable by a deterministic automaton all of whose states are accepting.

Maler and Pnueli \cite{13} provide an extension of the $L^*$ algorithm, using ultimately periodic words as examples, to the class of regular $\omega$-languages which are recognizable by both deterministic Büchi and deterministic co-Büchi automata. This is the subset for which the straightforward extension of right-congruence to infinite words gives a Myhill-Nerode characterization \cite{20}. Generalizing this to wider classes calls for finding a Myhill-Nerode characterization for larger classes of regular $\omega$-languages. This direction of research was taken in \cite{10, 14} and is one of the starting points of our work.

In fact the full class of regular $\omega$-languages can be learned using the result of Calbrix, Nivat and Podelski \cite{4}. They define for a given $\omega$-language $L$ the set $L_\Sigma = \{ u \Sigma^* v \mid u \Sigma^*, v \Sigma^+, uv^\omega \in L \}$ and show that $L_\Sigma$ is regular by constructing an NFA and a DFA accepting it. Since DFAs are canonical for regular languages, it follows that a DFA for $L_\Sigma$ is a canonical representation of $L$. Such a DFA can be learned by the $L^*$ algorithm provided the teacher’s counter examples are ultimately periodic words, given e.g. as a pair $(u, v)$ standing for $uv^\omega$ — a quite reasonable assumption that is common to the other works too. This DFA can be converted to a Büchi automaton recognizing it. This approach was studied and implemented by Farzan et al. \cite{7}. For a Büchi automaton with $m$ states, Calbrix et al. provide an upper bound of $2^m + 2^{2m^2 + m}$ on the size of a DFA for $L_\Sigma$. 
